BAB II

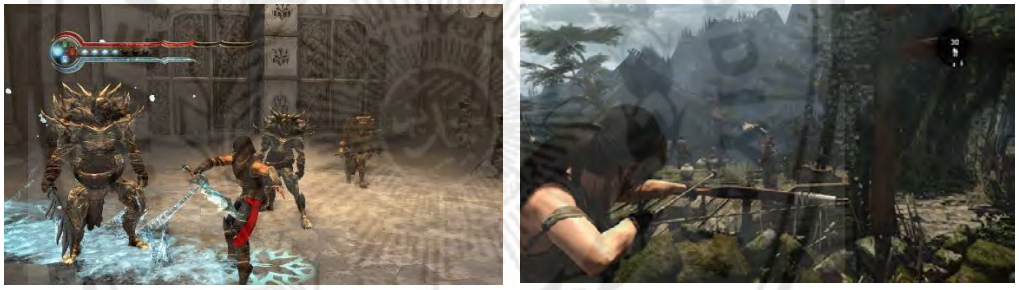
LANDASAN TEORI

2.1 Genre Game

Pada umumnya game dapat dikategorikan menjadi beberapa genre, seperti Action games, Adventure games, Platform games, RPG, board games, sport game, classic simulation games, fighting games, dan berbagai jenis genre lainnya yang tidak masuk dalam kategori yang sudah disebutkan.

2.1.1 Game Action (Game Aksi)

Game Action merupakan genre game yang memerlukan pemain untuk menggunakan refleks, akurasi, dan waktu yang tepat untuk menyelesaikan sebuah tantangan [9]. Contoh game action yang terkenal adalah: Doom, Tomb Raider dan Prince of Persia.



Gambar 2.1 Game Action Prince of Persia (kiri) dan Tomb Raider (kanan)

2.1.2 Game *Role Playing Games* (RPG Game)

Game RPG merupakan sebuah game di mana player memainkan suatu tokoh yang ada dalam game. Didalam game ini biasanya terdapat unsur seperti experience point, atau perkembangan karakter yang kita mainkan sehingga membuat karakter kita naik level dan semakin kuat. Unsur cerita dalam game RPG sangat kental. Ada yang akhir ceritanya bisa kita tentukan sendiri tergantung apa yang kita lakukan dalam game. Biasanya di game RPG terdapat juga sistem equipment, di mana untuk memperkuat karakter yang kita mainkan diperlukan kombinasi perlengkapan yang mempengaruhi dalam menjalankan game RPG [9]. Contoh game RPG adalah Ragnarog, Final Fantasi, dan Warcraft.

Gambar 2.2 Game RPG Final Fantasy (kiri) dan Dark Souls (kanan)

2.1.3 Game Adventure (Game Berpetualangan)

Game *Adventure* merupakan game yang biasanya memiliki 1 tokoh utama yang kita mainkan dan kita jalankan secara langsung dari awal sampai tamat mengikuti alur cerita [9]. Player game Adventure melakukan *quest* dimana player harus berinteraksi dengan NPC support untuk memecahkan puzzle dan mengungkap cerita game. Contoh *Adventure game* diantaranya adalah Rayman Origins, Crash Bandicoot, dan Bully.

Gambar 2.3 Game Adventure Bully (kiri) dan Crash Bandicoot (kanan)

2.1.4 Game Strategy (Strategi Game)

Game Strategy adalah game di mana kita sebagai pemain menjalankan berbagai unit-unit yang unik untuk memenangkan permainan tersebut. Gameplay nya biasanya kita mengatur unit atau pasukan untuk bertahan, menahan, bahkan mengalahkan musuh yang ada di game tersebut. Karena dalam permainan ini memerlukan biaya, pemain harus mengatur strategi dan mengalokasi sumber daya yang tepat sesuai apa yang dibutuhkan pemain [9]. Dalam permainan ini pemain tertantang untuk mengatur masalah jadwal, mengorganisir pertahanan dan penyerangan. Contoh permainan game strategi adalah *Civilisation Series* dan *Command and Conquer series.*

Gambar 2.4 Game Strategy Age of Empires (kiri) dan XCOM2 (kanan)

2.1.5 Game Sport (Olahraga)

Game *Sport* terdiri dari dua jenis. Yang pertama adalah olahraga individu (seperti olahraga golf, snowboarding, tinju, mengemudi, dll) dan yang kedua adalah olahraga tim (seperti sepak bola, basket, volly, rugby, dll). Dalam permainan individu, pemain mengendalikan karakter tunggal untuk bersaing baik berhadapan dengan karakter yang dikendalikan oleh komputer atau pemain yang dikendalikan oleh pemain lain [9]. Dalam olahraga tim, pemain memainkan peranan ganda yakni pembinaan sebuah peran pemain. Pemain dapat melatih dan mengambil keputusan strategis seperti memilih formasi dan pemain untuk tim. Contoh permainan Sport adalah FIFA dan Pro Evolution Soccer (PES) merupakan seri game yang popular untuk cabang sepakbola dan NBA untuk cabang basket.

Gambar 2.5 Game Sport FIFA (kiri) dan NBA (kanan)

2.1.6 Game Fighting (Bertarung)

Poin utama dalam permainan game ini adalah bertarung. Tujuan utama dari mengalahkan musuh dengan teknik dan jurus masing – masing karakter. *Player* bebas memilih karakter yang ingin digunakan [9]. Dalam permainan ini, pemain harus menguasai teknik bertarung seperti attack, blocking, counter-attack, dan skill lainnya [9]. Contoh game fighting adalah Tekken dan Marvel vs Capcom.

Gambar 2.6 Game Fighting Marvel vs Capcom (kiri) dan Tekken (kanan)

2.1.7 Game Simulation (Simulasi)

Game ini adalah game ini adalah replika dari dunia nyata. Game ini sangat membantu bagi seseorang yang belum pernah mencoba dalam dunia nyata misalkan mengendalikan pesawat terbang ataupun mengendarai mobil [9]. Game ini biasanya dibuat sedemikian realistis. Contoh game ini adalah Flight Simulator dan Driving Simulator.

Gambar 2.7 Game Simultion ETS (kiri) dan Flight Simulator 2015 (kanan)

2.2 *Non Playable Character* (NPC)

Non Playable Character (NPC) adalah karakter dalam permainan komputer modern yang tidak dapat dikendalikan oleh player. Karakter ini mempunyai kemampuan beradaptasi bereaksi sesuai dengan sesuai kondisinya. Tujuan dari pengembangan kecerdasan buatan pada NPC ini adalah membuat NPC sebisa mungkin lebih real atau nyata sehingga pola dan perilaku NPC dari satu waktu ke waktu bisa menstimulasikan apa yang semestinya player lakukan dalam situasi yang sama [9].

2.3 Behavior (Perilaku)

Perilaku atau behavior didefinisikan sebagai cara di mana seorang individu berperilaku atau bertindak. Ini merupakan cara seorang individu melakukan perilakunya sendiri. Perilaku harus dilihat dalam referensi untuk sebuah fenomena, suatu benda atau orang. Hal ini dapat dilihat dalam referensi untuk norma-norma masyarakat, atau cara di mana seseorang memperlakukan orang lain atau menangani benda. Oleh karena itu perilaku adalah cara seseorang bertindak terhadap orang-orang, masyarakat atau benda.

Hal ini dapat baik buruk atau baik. Hal ini dapat normal atau abnormal sesuai dengan norma-norma masyarakat [13].

2.4 Kecerdasan Buatan (Artificial Intilligence)

Kecerdasan buatan (Artificial Intilligence) merupakan Bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia [2].

2.5 Kecerdasan Buatan Dalam Game

Dalam dunia akademis, kecerdasan buatan dipelajari secara serius untuk meningkatkan kualitas hidup manusia. Para peneliti terus menerus mengembangkan teknik-teknik pada bidang ini untuk menghasilkan mesin yang semakin mengerti, dan memahami kebutuhan manusia. Dalam game berbasis kecerdasan buatan, ada 3 teknik yang diadaptasi dari bidang kecerdasan buatan untuk diterapkan pada game [14]. Beberapa diantaranya yaitu:

- **Pergerakan**

  Pola pergerakan merupakan cara yang sederhana untuk memberikan ilusi kecerdasan pada sebuah game. Game Galaga adalah contoh klasik penerapan pola pergerakan ini, dimana pesawat musuh dapat bergerak secara melingkat atau mengikuti pola garis lurus yang ditentukan. Contoh lain penerapan pola pergerakan adalah pada game first-person shooter yang menampilkan monster yang sedang berpatroli pada jalur tertentu, pada game simulasi pertempuran pesawat dimana pesawat musuh dapat melakukan manuver-manuver di udara yang menyulitkan kita mengejar, atau karakter-karakter non-player (figuran) seperti kambing yang sedang berjalan membutuhkan teknik pola pergerakan ini.

  Metode standar untuk menerapkan pola pergerakan adalah dengan cara menyimpan pola tersebut dalam suatu array. Array tersebut terdiri dari serangkaian koordinat atau perintah pergerakan dengan pola tertentu untuk mengontrol koordinat dari objek. Dengan metode ini, bisa didapatkan pola-pola pergerakan seperti melingkar, garis lurus, zig-zag atau bahkan kurva tak beraturan [14].

- **Mengambil Keputusan**

Pengambilan keputusan melibatkan karakter yang melakukan apa yang harus dilakukan selanjutnya. Biasanya, masing-masing karakter memiliki berbagai perilaku berbeda yang dapat mereka pilih untuk dilakukan: menyerang, berdiri diam, bersembunyi, menjelajahi, dan sebagainya. Perilaku adalah yang paling tepat pada setiap momen permainan. Perilaku yang dipilih kemudian dapat dieksekusi dengan menggunakan teknologi AI dan animasi gerakan.

Paling sederhana, karakter mungkin memiliki aturan yang sangat sederhana untuk memilih perilaku. Beberapa keputusan mungkin memerlukan gerakan AI untuk melaksanakannya [14].

- **Strategi**

Anda bisa pergi jauh dengan AI gerakan dan pengambilan keputusan AI, dan permainan berbasis threedimensional (3D) yang paling action hanya menggunakan dua elemen ini. Tapi untuk mengkoordinasikan seluruh tim, beberapa AI strategis dibutuhkan. Strategi mengacu pada pendekatan keseluruhan yang digunakan oleh sekelompok karakter. Dalam kategori ini algoritma AI yang tidak hanya mengendalikan satu karakter, namun mempengaruhi perilaku seluruh karakter. Setiap karakter dalam kelompok mungkin (dan biasanya akan) memiliki algoritma pengambilan keputusan dan pergerakan sendiri, namun secara keseluruhan pengambilan keputusan mereka akan dipengaruhi oleh strategi kelompok [14].

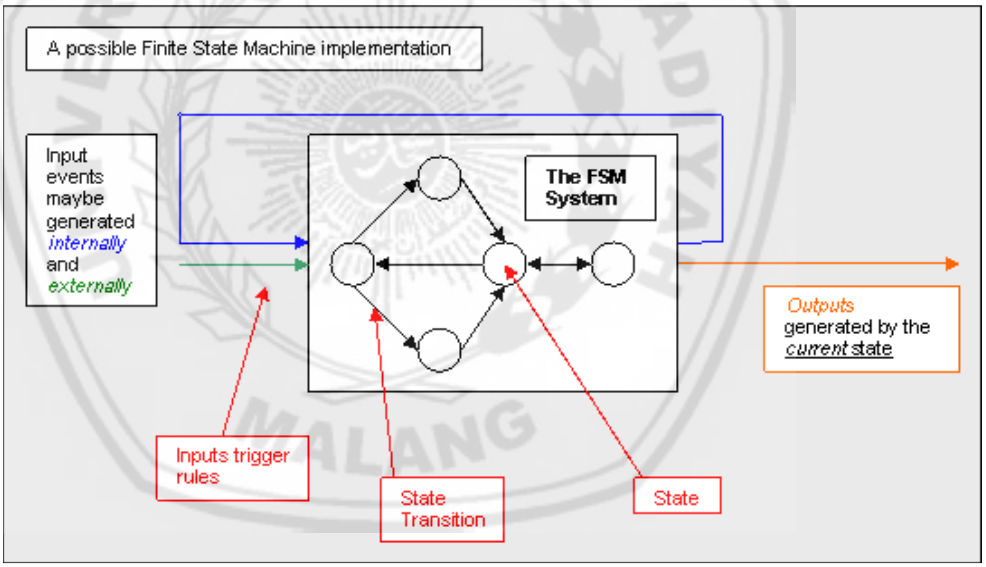
2.6 Finite State Machine (FSM)

Finite State Machine (FSM) adalah suatu model pemecahan *behavior* objek berdasarkan statenya. Sistem kerja FSM dengan menggunakan tiga hal berikut: *State* (Keadaan), *Event* (Kejadian) dan *Action* (Aksi) [11]. Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri (misal: interupsi timer). Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh system ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relative komplek.

Berdasarkan sifatnya, metode FSM ini sangat cocok digunakan sebagai basis perancangan perangkat lunak pengendalian yang bersifat reaktif dan real time. Selain untuk bidang pengendalian, penggunaan metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak game, aplikasi web, dan sebagainya [11].

State adalah keadaan objek saat ini dan *transition* adalah suatu acuan kondisi objek yang dilakukan agar bisa berpindah dari suatu *state* ke *state* yang lain. Dari gambar 2.8 dapat dijelaskan bahwa komponen utama FSM adalah:

1. *State* mendefinisikan perilaku dan bisa menghasilkan aksi
2. Transisi *state* adalah perpindahan dari satu *state* ke *state* yang lain.
3. Kondisi atau aturan yang harus terpenuhi supaya ada transisi *state.*
4. *Event* atau kejadian yang merupakan *input* yang dipicu oleh aturan dan mengacu ke transisi *state.*



Gambar 2.8 Contoh FSM sederhana [11]

FSM digambarkan sebagai jaringan semantik yang merepresentasikan maksud dan hubungan dengan menggunakan kata. State dilambangkan dengan lingkaran. Sedangkan transition disimbolkan dengan anak panah dengan arah tertentu. Tiap lingkaran dan anak panah memiliki nama masing – masing yang menunjukkan status *state* atau transisi [12]. FSM sendiri mempunyai keunggulan dan kekurangannya, antara lain:

Keunggulan FSM

1. Sederhana sehingga mudah diimplementasikan bagi orang awam.
2. Dapat diprediksi, transisi state mudah diprediksi sehingga memudahkan untuk melakukan uji coba.
3. Komputasi ringan.
4. Relatif fleksible sehingga mudah untuk digabungkan dengan teknik yang lain.
5. Merupakan metode AI lama yang bisadigunakan pada berbagai sistem dan telah teruji sebagai teknik pengembangan AI.
6. Mudah ditransfer dari konsep abstrak menjadi kode program komputasi.

Kekurangan FSM

1. Karena FSM mudah diprediksi, maka implementasi pada game kurang disukai.
2. Pada system yang lebih besar, implementasi FSM menjadi lebih sulit karena system menjadi lebih kompleks. Transisi state yang banyak menyebabkan faktor kerumitan jika mengikuti alurnya.
3. Sebaiknya hanya digunakkan pada sistem dimana sifat sistem bisa didekomposisi menjadi state terpisah dengan definisi kondisi yang jelas untuk transisi state. Dalam artian semua state, trarnsisi dan kondisi harus diketahui sebelumnya dan terdifinisikan dengan baik.
4. Kondisi untuk setiap state adalah tetap.

2.7 Fuzzy

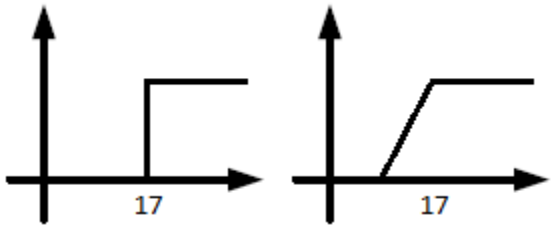
Fuzzy dalam bahasa Inggris dapat diartikan sebagai tidak jelas atau kabur.

2.7.1 Logika Fuzzy

Dalam bahasa Inggris, *fuzzy* mempunyai arti kabur atau tidak jelas. Sedangkan Logika merupakan studi penalaran. Pada teori logika yang biasa, logika dinyatakan dengan benar atau salah. Namun dalam kehidupan sehari-hari, sering ditemukan kasus yang tidak bisa dinyatakan sebagai benar atau salah, tapi harus dinyatakan dengan hampir benar, agak benar atau semacamnya. Jadi, logika *fuzzy* adalah logika yang kabur, atau mengandung unsur ketidakpastian [5].

Pada logika biasa yaitu logika tegas, kita hanya mengenal dua nilai yaitu salah atau benar, 0 atau 1. Sedangkan logika *fuzzy* mengenal nilai antara benar dan salah. Kebenaran dalam logika *fuzzy* dapat dinyatakan dalam derajat kebenaran yang nilainya antara 0 sampai

1. Misalnya dalam kehidupan sehari-hari, dewasa didefinisikan dengan berusia 17 tahun ke atas. Jika menggunakan logika tegas, seseorang yang berusia 17 tahun kurang 1 hari akan didefinisikan sebagai tidak dewasa. Namun dalam logika *fuzzy*, orang tersebut dapat dinyatakan dengan hampir dewasa.



Gambar 2.9 Logika tegas (kiri) dan logika fuzzy (kanan) [5]

2.7.2 Sejarah Logika Fuzzy

Logika fuzzy pertama kali dikembangkan oleh Prof. Lotfi A. Zadeh, seorang peneliti dari Universitas California, pada tahun 1960-an. Logika fuzzy dikembangkan dari teori himpunan fuzzy [5].

2.7.3 Himpunan Fuzzy

Himpunan fuzzy adalah pengelompokan sesuatu berdasarkan variabel bahasa yang dinyatakan dengan fungsi keanggotaan. Keanggotaan suatu nilai pada himpunan dinyatakan dengan derajat keanggotaan yang nilainya antara 0.0 sampai 1.0 [5].

Himpunan fuzzy didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval (0,1). Nilai keanggotaannya menunjukkan bahwa suatu item tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah [5].

2.7.4 Kelebihan dan Kekurangan Logika Fuzzy

Logika fuzzy memiliki beberapa keunggulan, antara lain sebagai berikut.

1. Konsep logika fuzzy mudah dimengerti.
2. Logika Fuzzy sangat fleksibel.
3. Logika Fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika Fuzzy dapat memodelkan fungsi-fungsi non linear yang kompleks.
5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.

6. Logika Fuzzy dapat bekerja sama dengan teknik-teknik kendali secara konvensional.

Sementara itu, dalam pengaplikasiannya, logika *fuzzy* juga memiliki beberapa kelebihan, antara lain sebagai berikut.

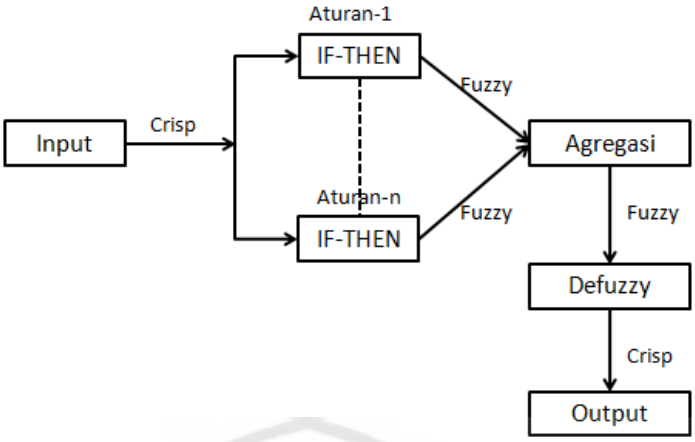
1. Daya gunannya dianggap lebih baik daripada teknik kendali yang pernah ada.
2. Pengendali fuzzy terkenal karena keandalannya.
3. Mudah diperbaiki.
4. Pengendali fuzzy memberikan pengendalian yang sangat baik dibandingkan teknik yang lain.

Selain itu, logika fuzzy juga memiliki kekurangan, terutama dalam penerapannya. Kekurangan-kekurangan tersebut antara lain:

1. Para ilmuwan generasi sebelumnya dan sekarang banyak yang tidak mengenal teori kendali fuzzy, meskipun secara teknik praktis mereka memiliki pengalaman untuk menggunakan teknologi dan perkakas kontrol yang sudah ada.
2. Belum banyak terdapat kursus/balai pendidikan dan buku-buku teks yang menjangkau setiap tingkat pendidikan (undergraduate, postgraduate, dan on site training).
3. Hingga kini belum ada pengetahuan sistematik yang baku dan seragam tentang metodologi pemecahan problema kendali menggunakan pengendali fuzzy.

2.7.5 Sistem Inferensi Fuzzy

Sistem Inferensi Fuzzy (Fuzzy Inference System atau FIS) merupakan penarikan kesimpulan dari sekumpulan kaidah fuzzy, aturan fuzzy berbentuk if- THEN, dan penalaran fuzzy secara garis besar, digram blok proses inferensi fuzzy terlihat pada Gambar 2.10.

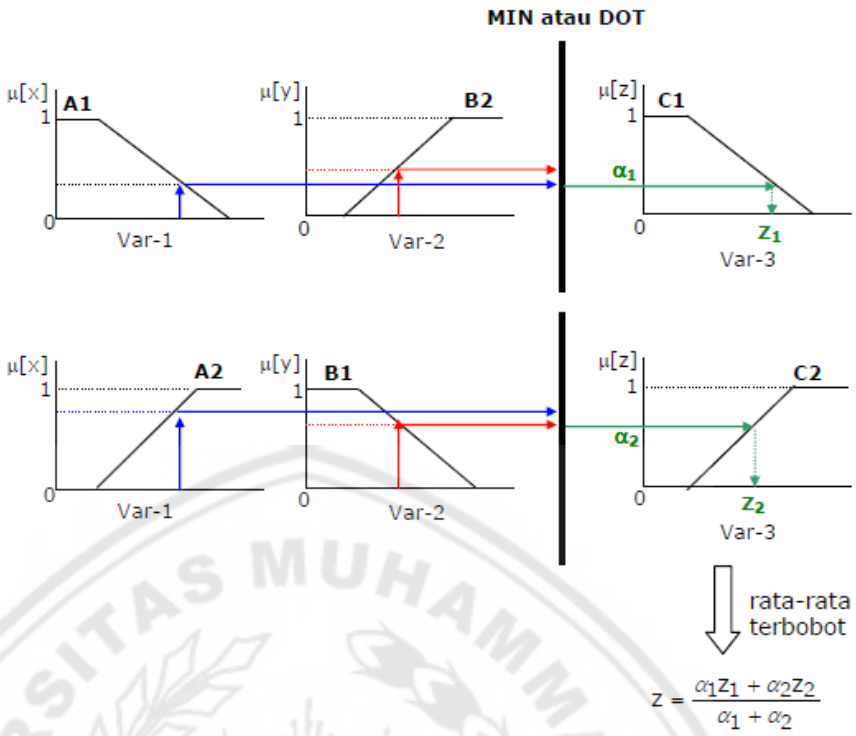


Gambar 2.10 Diagram Blok Sistem Inferensi Fuzzy [16]

Sistem inferensi fuzzy menerima input crisp. Input ini kemudian dikirim ke basis pengetahuan yang berisi n aturan fuzzy dalam bentuk IF – THEN. Fire strength akan dicari pada setiap aturan. Apabila jumlah aturan lebih dari satu, maka akan dilakukan agregasi dari semua aturan. Selanjutnya, pada hasil agregasi akan dilakukan defuzzy untuk mendapatkan nilai crisp sebagai output sistem [16].

2.7.5.1 Metode Tsukamoto

Setiap konsekuen pada aturan berbentuk IF-THEN direpresentasikan dengan suatu himpunan Fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasil, output tiap-tiap aturan diberikan secara tegas berdasar α-predikat (fire strenght) [16].



Gambar 2.11 Inferensi dengan Menggunakan Metode Tsukamoto [16]

CONTOH KASUS 2.1:

Sebuah perusahaan makanan kaleng akan memproduksi makanan jenis ABC. Dari data 1 bulan terakhir, PERMINTAAN TERBESAR mencapai 5000 kemasan/hari, dan PERMINTAAN TERKECIL 1000 kemasan/hari. PERSEDIAAN TERBANYAK digudang sampai 600 kemasan/hari, dan PERSEDIAAN TERKECIL mencapai 100 kemasan/hari. Dengan segala keterbatasan kemampuan PRODUKSI TERBANYAK adalah 7000 kemasan/hari, dan agar efisien PRODUKSI TERKECIL adalah 2000 kemasan/hari. Dalam produksi perusahaan menggunakan aturan:

R1 : JIKA permintaan TURUN dan persediaan BANYAK maka produksi BERKURANG

R2 : JIKA permintaan TURUN dan persediaan SEDIKIT maka produksi BERKURANG

R3 : JIKA permintaan NAIK dan persediaan BANYAK maka produksi BERTAMBAH

R4 : JIKA permintaan NAIK dan persediaan SEDIKIT maka produksi BERTAMBAH

Berapa harus diproduksi jika PERMINTAAN 4000 kemasan dan PERSEDIAAN 300 kemasan?.

SOLUSI:

Terdapat 3 variabel fuzzy yaitu (1) permintaan, (2) persediaan, dan (3) produksi

- **PERMINTAAN**

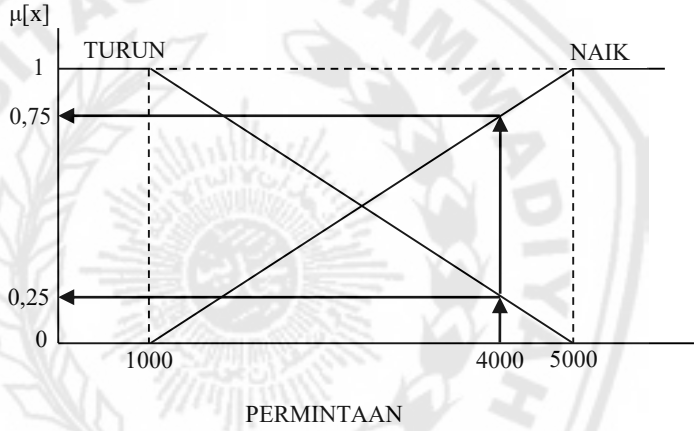
  Terdiri dari 2 himpunan fuzzy, yaitu (1) TURUN, dan (2) NAIK

  Diketahui :

  Permintaan terendah adalah 1000 kemasan/hari

  Permintaan tertinggi adalah 5000 kemasan/hari

  Permintaan permasalahan = 4000 kemasan



Gambar 2.12 Fungsi Keanggotaan Variabel Permintaan Pada Contoh Kasus 2.1 [16]

$$\mu_{permintaan-turun}[x] \begin{cases} 1 & x \leq 1000 \\ \frac{5000-x}{4000} & 1000 \leq x \leq 5000 \\ 0 & x \geq 5000 \end{cases} \quad \text{.....(1)}$$

$$\mu_{permintaan-naik}[x] \begin{cases} 0 & x \leq 1000 \\ \frac{x-1000}{4000} & 1000 \leq x \leq 5000 \\ 1 & x \geq 5000 \end{cases} \quad \text{.....(2)}$$

Rumus 1 digunakan untuk menghitung permintaan turun yang berkisar antara 1000 sampai 5000. Jika angka permintaan mencapai kurang dari 1000, maka hasil perhitungan adalah 1. Jika angka permintaan mencapai 1000 sampai 5000,

maka akan dihitung dengan menggunakan rumus dan jika angka permintaan mencapai lebih dari 5000 maka hasil perhitungan adalah 0.

Rumus 2 digunakan untuk menghitung permintaan naik yang berkisar antara 1000 sampai 5000. Jika angka permintaan mencapai kurang dari 1000, maka hasil perhitungan adalah 0. Jika angka permintaan mencapai 1000 sampai 5000, maka akan dihitung dengan menggunakan rumus dan jika angka permintaan mencapai lebih dari 5000 maka hasil perhitungan adalah 1.

- **PERSEDIAAN**

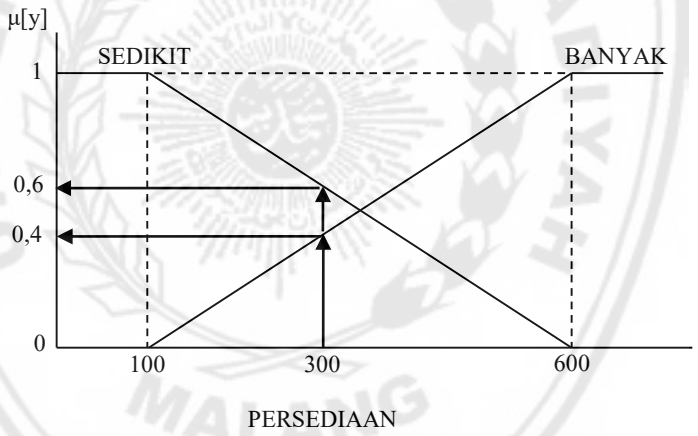
Terdiri dari 2 himpunan fuzzy, yaitu (1) SEDIKIT, dan (2) BANYAK

Diketahui :

Persediaan terendah adalah 100 kemasan/hari

Persediaan tertinggi adalah 600 kemasan/hari

Persediaan permasalahan = 300 kemasan



Gambar 2.13 Fungsi Keanggotaan Variabel Persediaan Pada Contoh Kasus 2.1 [16]

$$\mu_{persediaan-sedikit}[y] \begin{cases} 1 & y \leq 100 \\ \frac{600-y}{500} & 100 \leq y \leq 600 \\ 0 & y \geq 600 \end{cases} \quad \text{.....(3)}$$

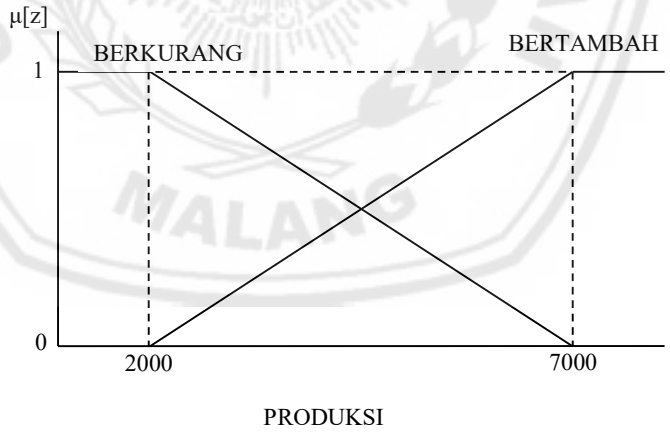
$$\mu_{persediaan-banyak}[y] \begin{cases} 0 & y \leq 100 \\ \frac{y-100}{500} & 100 \leq y \leq 600 \\ 1 & y \geq 600 \end{cases} \quad \text{.....(4)}$$

Rumus 3 digunakan untuk menghitung persediaan sedikit yang berkisar antara 100 sampai 600. Jika angka persediaan mencapai kurang dari 100, maka hasil perhitungan adalah 1. Jika angka persediaan mencapai 100 sampai 600, maka akan dihitung dengan menggunakan rumus dan jika angka persediaan mencapai lebih dari 600 maka hasil perhitungan adalah 0.

Rumus 4 digunakan untuk menghitung persediaan banyak yang berkisar antara 100 sampai 600. Jika angka persediaan mencapai kurang dari 100, maka hasil perhitungan adalah 0. Jika angka persediaan mencapai 100 sampai 600, maka akan dihitung dengan menggunakan rumus dan jika angka persediaan mencapai lebih dari 600 maka hasil perhitungan adalah 1.

- **PRODUKSI**

Terdiri dari 2 himpunan fuzzy, yaitu (1) BERKURANG, dan (2) BERTAMBAH
Diketahui:
Produksi terendah adalah 2000 kemasan/hari
Produksi tertinggi adalah 7000 kemasan/hari
Produksi permasalahan = ditanyakan ?? kemasan



Gambar 2.14 Fungsi Keanggotaan Variabel Produksi Pada Contoh Kasus 2.1[16]

$$\mu_{produksi-berkurang}[z] \begin{cases} 1 & z \leq 2000 \\ \frac{7000-z}{5000}, & 2000 \leq z \leq 7000 \\ 0 & z \geq 7000 \end{cases} \quad \text{.....(5)}$$

$$\mu_{produksi-bertambah}[z] \begin{cases} 0 & z \leq 2000 \\ \frac{z-2000}{5000}, & 2000 \leq z \leq 7000 \\ 1 & z \geq 7000 \end{cases} \quad \text{.....(6)}$$

Rumus 5 digunakan untuk menghitung produksi berkurang yang berkisar antara 2000 sampai 7000. Jika angka produksi mencapai kurang dari 2000, maka hasil perhitungan adalah 1. Jika angka produksi mencapai 2000 sampai 7000, maka akan dihitung dengan menggunakan rumus dan jika angka produksi mencapai lebih dari 7000 maka hasil perhitungan adalah 0.

Rumus 6 digunakan untuk menghitung produksi bertambah yang berkisar antara 2000 sampai 7000. Jika angka produksi mencapai kurang dari 2000, maka hasil perhitungan adalah 0. Jika angka produksi mencapai 2000 sampai 7000, maka akan dihitung dengan menggunakan rumus dan jika angka produksi mencapai lebih dari 7000 maka hasil perhitungan adalah 1.

Cari Nilai Produksi Z, dengan fungsi implikasi MIN

- **Permintaan x**

Tabel 2.1 Perhitungan *Keanggotaan Variabel Permintaan Pada Contoh Kasus 2.1[16]*

Fungsi keanggotaan TURUN : $\mu_{permintaan-turun}[x] \begin{cases} 1 & x \leq 1000 \\ \frac{5000-x}{4000}, & 1000 \leq x \leq 5000 \\ 0 & x \geq 5000 \end{cases}$	**Permintaan = 4000** $\mu_{permintaan-turun}[4000] = \frac{5000-4000}{4000}$ $= 0{,}25$
Fungsi keanggotaan NAIK : $\mu_{permintaan-naik}[x] \begin{cases} 0 & x \leq 1000 \\ \frac{x-1000}{4000}, & 1000 \leq x \leq 5000 \\ 1 & x \geq 5000 \end{cases}$	**Permintaan = 4000** $\mu_{permintaan-naik}[4000] = \frac{4000-1000}{4000}$ $= 0{,}75$

- **Persediaan y**

Tabel 2.2 Perhitungan *Keanggotaan Variabel Persediaan Pada Contoh Kasus 2.1[16]*

Fungsi keanggotaan SEDIKIT : $\mu_{persediaan-sedikit[y]} \begin{cases} 1 & y \leq 100 \\ \frac{600-y}{500} & 100 \leq y \leq 600 \\ 0 & y \geq 600 \end{cases}$	**Persediaan = 300** $\mu_{persediaan-sedikit[300]} = \frac{600-300}{500}$ $= 0{,}6$
Fungsi keanggotaan BANYAK : $\mu_{persediaan-banyak[y]} \begin{cases} 0 & y \leq 100 \\ \frac{y-100}{500} & 100 \leq y \leq 600 \\ 1 & y \geq 600 \end{cases}$	**Permintaan = 300** $\mu_{persediaan-banyak[300]} = \frac{300-600}{500}$ $= 0{,}4$

- **Mencari Produksi z**

R1 : JIKA permintaan TURUN dan persediaan BANYAK maka produksi BERKURANG

$$\alpha_{-predikat1} = \mu_{permintaan-turun} \cap \mu_{persediaan-banyak}$$
$$= \min(\mu_{permintaan-turun}[4000] \cap \mu_{persediaan-banyak}[300]) \quad .....(7)$$
$$= \min(0{,}25; 0{,}4)$$
$$= 0{,}25$$

$$\mu_{produksi-berkurang[z]} \begin{cases} 1 & z \leq 2000 \\ \frac{7000-z}{5000}, & 2000 \leq z \leq 7000 \\ 0 & z \geq 7000 \end{cases} \quad .....(8)$$

$$\frac{7000-z1}{5000} = 0{,}25 \quad \rightarrow z1 = 5750 \quad .....(9)$$

Rumus 7 menjelaskan perhitungan 1 aturan kemudian dari masing – masing variabel diambil nilai terkecilnya. Kemudian pada rumus 8 menjelaskan rumus untuk menghitung produksi berkurang yang berkisar antara 2000 sampai 7000. Jika angka produksi mencapai kurang dari 2000, maka hasil perhitungan adalah 1. Jika angka produksi mencapai 2000 sampai 7000, maka akan dihitung dengan menggunakan rumus dan jika angka produksi mencapai lebih dari 7000 maka hasil perhitungan adalah 0. Pada rumus 9 mencari nilai z1 dengan cara nilai terkecil dari

variable dikalikan dengan 5000, kemudian nilai 7000 akan dikurangi oleh hasil perkalian 5000 dengan nilai variable terkecil.

R2 : JIKA permintaan TURUN dan persediaan SEDIKIT maka produksi BERKURANG

$$\alpha_{-predikat2} = \mu_{permintaan-turun} \cap \mu_{persediaan-sedikit}$$
$$= \min(\mu_{permintaan-turun}[4000] \cap \mu_{persediaan-sedikit}[300]) \quad \text{.....(10)}$$
$$= \min(0{,}25; 0{,}6)$$
$$= 0{,}25$$

$$\boldsymbol{\mu_{produksi-berkurang}[z]} \begin{cases} 1 & z \leq 2000 \\ \frac{7000-z}{5000}, & 2000 \leq z \leq 7000 \\ 0 & z \geq 7000 \end{cases} \quad \text{.....(11)}$$

$$\frac{7000-z2}{5000} = 0{,}25 \quad \rightarrow z2 = 5750 \quad \text{.....(12)}$$

Rumus 10 menjelaskan perhitungan 1 aturan kemudian dari masing – masing variabel diambil nilai terkecilnya. Pada rumus 11 menjelaskan rumus untuk menghitung produksi berkurang yang berkisar antara 2000 sampai 7000. Jika angka produksi mencapai kurang dari 2000, maka hasil perhitungan adalah 1. Jika angka p produksi mencapai 2000 sampai 7000, maka akan dihitung dengan menggunakan rumus dan jika angka produksi mencapai lebih dari 7000 maka hasil perhitungan adalah 0. Pada rumus 12 mencari nilai z2 dengan cara nilai terkecil dari variable dikalikan dengan 5000, kemudian nilai 7000 akan dikurangi oleh hasil perkalian 5000 dengan nilai variable terkecil.

R3 : JIKA permintaan NAIK dan persediaan BANYAK maka produksi BERTAMBAH

$$\alpha_{-predikat3} = \mu_{permintaan-naik} \cap \mu_{persediaan-banyak}$$
$$= \min(\mu_{permintaan-naik}[4000] \cap \mu_{persediaan-banyak}[300]) \quad \text{.....(13)}$$
$$= \min(0{,}75; 0{,}4)$$
$$= 0{,}4$$

$$\boldsymbol{\mu_{produksi-bertambah}[z]} \begin{cases} 0 & z \leq 2000 \\ \frac{z-2000}{5000}, & 2000 \leq z \leq 7000 \\ 1 & z \geq 7000 \end{cases} \quad \text{.....(14)}$$

$$\frac{z3-2000}{5000} = 0{,}4 \rightarrow z3 = 4000 \quad \text{.....(15)}$$

Rumus 13 menjelaskan perhitungan 1 aturan kemudian dari masing – masing variabel diambil nilai terkecilnya. Rumus 14 menjelaskan rumus untuk untuk menghitung produksi bertambah yang berkisar antara 2000 sampai 7000. Jika angka produksi mencapai kurang dari 2000, maka hasil perhitungan adalah 0. Jika angka produksi mencapai 2000 sampai 7000, maka akan dihitung dengan menggunakan rumus dan jika angka produksi mencapai lebih dari 7000 maka hasil perhitungan adalah 1. Pada rumus 15 mencari nilai z3 dengan cara nilai terkecil dari variable dikalikan dengan 5000, kemudian nilai 2000 akan dijumlah oleh hasil perkalian 5000 dengan nilai variable terkecil.

R4 : JIKA permintaan NAIK dan persediaan SEDIKIT maka produksi BERTAMBAH

$$\alpha_{-predikat4} = \mu_{permintaan-naik} \cap \mu_{persediaan-sedikit}$$
$$= \min(\mu_{permintaan-naik}[4000] \cap \mu_{persediaan-sedikit}[300]) \quad \text{.....(16)}$$
$$= \min(0{,}75; 0{,}6)$$
$$= 0{,}6$$

$$\boldsymbol{\mu_{produksi-bertambah}[z]} \begin{cases} 0 & z \leq 2000 \\ \frac{z-2000}{5000}, & 2000 \leq z \leq 7000 \\ 1 & z \geq 7000 \end{cases} \quad \text{.....(17)}$$

$$\frac{z4-2000}{5000} = 0{,}6 \rightarrow z4 = 5000 \quad \text{.....(18)}$$

Rumus 16 menjelaskan perhitungan 1 aturan kemudian dari masing – masing variabel diambil nilai terkecilnya. Rumus 17 menjelaskan rumus untuk untuk menghitung produksi bertambah yang berkisar antara 2000 sampai 7000. Jika angka produksi mencapai kurang dari 2000, maka hasil perhitungan adalah 0. Jika angka produksi mencapai 2000 sampai 7000, maka akan dihitung dengan menggunakan rumus dan jika angka produksi mencapai lebih dari 7000 maka hasil perhitungan adalah 1. Pada rumus 18 mencari nilai z4 dengan cara nilai terkecil dari variable dikalikan dengan 5000, kemudian nilai 2000 akan dijumlah oleh hasil perkalian 5000 dengan nilai variable terkecil.

Hitung z sebagai berikut :

$$z = \frac{\alpha_{-predikat1} * z1 + \alpha_{-predikat2} * z2 + \alpha_{-predikat3} * z3 + \alpha_{-predikat4} * z4}{\alpha_{-predikat1} + \alpha_{-predikat2} + \alpha_{-predikat3} + \alpha_{-predikat4}} \quad \text{.....(19)}$$

$$z = \frac{0{,}25*5750 + 0{,}25*5750 + 0{,}4*4000 + 0{,}6*5000}{0{,}25 + 0{,}25 + 0{,}4 + 0{,}6} \quad \text{.....(20)}$$

$$z = \frac{7475}{1,5} = 4983 \qquad \text{.....(21)}$$

Rumus 19, 20 dan 21 menjelaskan perhitungan untuk mencari hasil keseluruhan dari jumlah makanan yang harus diproduksi dengan melibatkan nilai terkecil pada 4 variable dan hasil perhitungan dari 4 aturan.

2.7.5.2 Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama metode Max-Min. metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 (Kusuma Dewi, 2003). Untuk mendapatkan output diperlukan beberapa tahapan, antara lain:

1. **Pembentukan Himpunan Fuzzy**

   Pada Metode Mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy.

2. **Aplikasi fungsi implikasi**

   Pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min.

3. **Komposisi Aturan**

   Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan kolerasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem fuzzy, yaitu max, additive dan probabilistik OR (probor).

a) Metode Max (Maximum)

Metode Max (Maximum) mengambil solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy, dan mengapilasikannya ke output dengan menggunakan operator OR (unionMetode Additive (Sum). Metode Additive (Sum) mengambil solusi himpunan fuzzy diperoleh dengan cara melakukan bounded-sum terhadap semua output daerah fuzzy.

b) Metode Probabilistik OR (probor)

Metode Probabilitik OR (probor) mengambil solusi himpunan fuzzy diperoleh dengan cara melakukan product terhadap semua output daerah fuzzy.

c) Penegasan (defuzzy)

Input dari proses defuzzyfikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut

2.7.5.3 Metode Sugeno

Secara umum menyerupai metode MAMDANI, akan tetapi output/konsekuen berupa konstanta atau persamaan linear [16].

2.8 Dynamic Game Balancing

Dynamic Game Balancing adalah proses yang harus dipenuhi setidaknya untuk 3 ketentuan. Pertama game tersebut perlu secepat mungkin mengenal dan menyesuaikan dirinya dengan tingkat dasar pemain manusianya, yang akan berbeda secara luas dari pemula hingga ahli. Kedua, game tersebut harus mengikuti sedekat dan secepat mungkin dengan evolusi dan kemunduran dari kinerja pemain. Ketiga, dalam mengadaptasi dirinya sendiri, perilaku dari game tersebut harus tetap terpercaya sejak pengguna tidak dimaksudkan untuk memahami bahwa komputer bermain dengan tangan virtual yang terikat ke belakang (misal, menjalankan aksi yang jelas-jelas mengalahkan diri sendiri) serta karakter permainan yang diharapkan oleh pengguna [15].